C++ ASSIGNMENT

1.Ques :Given a sorted array of n elements and a target ‘x’. Find the last occurrence of ‘x’ in the array. If ‘x’ does not exist return -1.
Input 1: arr[] = {1,2,3,3,4,4,4,5} , x = 4
Output 1: 6

Ans:
```
#include<iostream>
#include<vector>
using namespace std;
int main(){
    int arr[]={1,2,2,3,3,3,3,4,4,5};
    int n=10;
    int x=3;
    int lo=0;
    int hi=n-1;
    bool flag=false;
    while(lo<=hi){
        int mid=lo+(hi-lo)/2;
        if(arr[mid]==x){
            if(arr[mid+1]!=x){
                cout<<mid;
                flag=true;
                break;
            }
            else lo=mid+1;
        }
        else if(arr[mid]<x) lo=mid+1;
        else hi=mid-1;
  }
  if(flag==false) cout<<-1;
}
```

2.Ques :Given a sorted binary array, efficiently count the total number of 1’s in it.

Input 1 : a = [0,0,0,0,1,1]

Output 1: 2

Ans:

```
#include<iostream>
#include<vector>
using namespace std;
int main(){
    int arr[]={0,0,0,0};
    int n=8;
    int lo=0;
    int hi=n-1;
    int f=-1;
    while(lo<=hi){
        int mid=lo+(hi-lo)/2;
        if(mid==0){
            f=mid;
            break;
        }
        if(arr[mid]==1){
            if(arr[mid-1]!=1){
                f=mid;
                break;
            }
            else hi=mid-1;
        }
        else if(arr[mid]<1) lo=mid+1;
        else hi=mid-1;
  }
  if(f==-1) cout<<0;
  else cout<<n-f;
}
```

3.Ques :Given a matrix having 0-1 only where each row is sorted in increasing order, find the row with the maximum number of 1’s.

Input matrix : 0 1 1 1
0 0 1 1
1 1 1 1 // this row has maximum 1s
0 0 0 0
Output: 2

Ans:

```
#include<iostream>
#include<vector>
using namespace std;
int main(){
    int n=3;
    int m=4;
    int arr[n][m]={{0,0,0,0},{0,1,1,1},{1,1,1,1}};
    int count=0;
    int maxone=-1;
    int maxrow=-1;
    int f=-1;
    bool flag=false;
    for(int i=0;i<n;i++){
        int lo=0;
        int hi=m-1;
        while(lo<=hi){
        int mid=lo+(hi-lo)/2;
        if(mid==0){
            f=mid;
            flag=true;
            break;
        }
        if(arr[i][mid]==1){
            if(arr[i][mid-1]!=1){
                f=mid;
                break;
            }
            else hi=mid-1;
        }
        else if(arr[i][mid]<1) lo=mid+1;
        else hi=mid-1;
  }
    if(flag==true) count=m-f;
  if(maxone<count){
    maxone=count;
    maxrow=i;
    }
    }
    cout<<maxone<<" "<<maxrow;
}
```

4.Ques:Given an array of integers nums containing n + 1 integers where each integer is in the range [1, n] inclusive in sorted order.

There is only one repeated number in nums, return this repeated number.

Input 1: arr[] = {1,2,3,3,4}

Output 1: 3

Input 2: arr[] = {1,2,2,3,4,5}

Output 2: 2

Ans:

```
#include<iostream>
#include<vector>
using namespace std;
int main(){
        int arr[]={1,2,3,3,4,5};
        int n=6;
        int lo=0;
        int hi=n-1;
        bool flag=false;
        while(lo<=hi&&lo<n){
                int mid=lo+(hi-lo)/2;
                if(arr[mid]==arr[mid-1]||arr[mid]==arr[mid+1]){
                        cout<<arr[mid];
                        flag=true;
                        break;
                }
                else lo=mid+1;
   }
   if(flag==false) cout<<-1;
}
```

5.Ques:Given a number ‘n’. Predict whether ‘n’ is a valid perfect square or not.

Input 1: n = 36

Output 1: yes

Input 2: n = 45

Output 2: no

Array that contains only positive elements.

Ans:

```
#include<iostream>
```

```
#include<vector>
using namespace std;
int main(){
    int n;
    cout<<"Enter the number: ";
    cin>>n;
    int lo=0;
    int hi=n;
    bool flag=false;
    while(lo<=hi){
        int mid=lo+(hi-lo)/2;
        if(mid*mid==n){
            flag=true;
            break;
        }
        else if(mid*mid>n) hi=mid-1;
        else lo=mid+1;
  }
  if(flag==false) cout<<"NO";
  else cout<<"YES";
}
```

6.Ques:You have n coins and you want to build a staircase with these coins. The staircase consists of k rows where the ith row has exactly i coins. The last row of the staircase may be incomplete. Given the integer n, return the number of complete rows of the staircase you will build.

Example 1:

Input: n = 5

Output: 2

Explanation: Because the 3rd row is incomplete, we return 2.

Example 2:
Input: n = 8
Output: 3
Explanation: Because the 4th row is incomplete, we return 3.

Ans:

```
#include<iostream>
#include<vector>
using namespace std;
int main(){
        int n;
        cout<<"Enter no.of coins:";
        cin>>n;
        int lo=1;
        int hi=n;
        bool flag=false;
        while(lo<=hi){
                int m=lo+(hi-lo)/2;
                int k=m*(m+1)/2;
                if(k==n){
                        flag=true;
                        cout<<m;
                        break;
                }
                if(k>n) hi=m-1;
                else lo=m+1;
}
 if(flag==false) cout<<hi;
}
```